This is a brief outline of some fundamental problems inherent in all e-voting systems, starting with a comment on the idea of open source code for such systems. I recommend a much simpler, more secure, voting mechanism that is widely used with an excellent track record.

Open Source Code for e-voting systems addresses an obvious problem with such systems. Namely that it is absurd to have the details of the basic method by which citizens control a democratic government concealed from them. But, even if the open source movement, which I believe is being powered by very fine, able, people with the best intentions, should succeed completely, there would remain a virtually limitless array of techniques for falsifying the results of elections using e-voting.

The political election process is a rare example of a data processing task that does not lend itself to implementation by computers. There is no feasible way to ensure that a particular instance of an e-voting system does not have clandestine features for corrupting the results. This is because the number of different hidden cheating techniques is bounded only by the ingenuity of the designers.

Some ingenious schemes have been proposed for building in features that would allow voters to check, after the election results are posted, to see if their votes have been listed. But there is ample evidence that post-election efforts to correct election outcomes seldom are effective. In a number of cases, voters have complained that their votes were not correctly recorded even on the polling booth screens, but no effective remedial action was taken, Even where there was an extensive post-election investigation of such a case--the Sarasota undervote--it was clear that there was no way to determine if deliberate cheating was the cause.

It is important to understand that, while source code is an obvious possible culprit, cheating via embedded hardware features, which are even harder to detect, is quite possible. The process whereby the source code is converted to object code (compilers, assemblers, loaders, etc.) are also potential sources of corruption. Most recently it has been demonstrated that the firmware associated with the BIOS (Basic Input/Output System), a component of every computer, is yet another tool available for cheating. (see http://threatpost.com/en_us/blogs/researchers-unveil-persistent-bios-attack-methods-031909)

Fortunately there is no need to put up with faith-based elections, i.e., a situation in which the integrity of our elections depends on the honesty and competence of an array of election officials, engineers and technicians. We can mark paper ballots by hand and have them counted by hand, in public, with the entire process monitored by representatives of competing political organizations. This approach is, in fact, widely used in several states and works just fine. Of course, regardless of the use or non-use of high technology, it remains critical that we pay attention to the voting and tabulation process, never trusting that unobserved technicians or government officials will do the right thing.

I have elaborated on the above contentions in a number of articles, which include references to other work.

1. The general case for hand counted paper ballots (HCPB) is made at http://www1.cs.columbia.edu/~unger/articles/manualCount.html

2. An analysis of problems with e-voting is at http://www1.cs.columbia.edu/~unger/articles/e-voting1-11-07.html

3. The Sarasota case is analyzed at http://www1.cs.columbia.edu/~unger/articles/sarasota5-2-07.html

Stephen H. Unger
Professor Emeritus
Computer Science and Electrical Engineering Columbia University
....